

Prática 03 - AED 2023.1

Nós e Listas Encadeadas - Programação Estruturada

Abril de 2023

1 Objetivos

Ao final dessa tarefa o aluno será capaz de:

- **Lembrar** recursos avançados do Python e do C++, **recordando-os** ao construir algoritmos e programas;
- **Entender** recursos avançados e sua sintaxe no C++ e Python, **interpretando** o código-fonte de programas;
- **Aplicar** técnicas de programação avançada em problemas simples, **executando** programações de algoritmos usando tais recursos;
- **Criar** soluções computacionais, **construindo-as** a partir dos conceitos aprendidos;

2 Descrição da Atividade

O Nó é a unidade básica da qual são constituídas diversos Tipos Abstratos de Dados como Listas Encadeadas, Árvores e Grafos. Um nó armazena um valor(chave) que pode ser qualquer dado e um ponteiro para outro(s) nó(s). A Figura 1 ilustra esta unidade.

2.1 Parte I

O nó é um agregado heterogêneo de dados, portanto, pode ser implantado computacionalmente utilizando estruturas e dicionários. Com base nessa informação faça o que se pede:

1. Implementar em linguagem C++, a estrutura `no`. Os membros da estrutura são

- chave: dado do tipo `int`;
- prox: dado do tipo ponteiro para o tipo `no`;

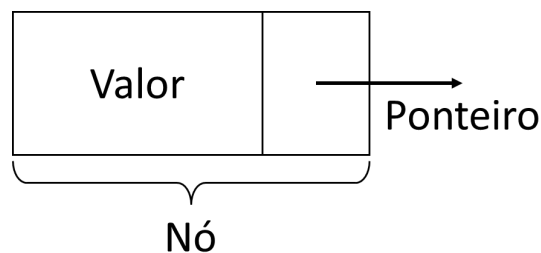


Figura 1: Representação visual de um nó

2. Implementar em linguagem Python, a representação de um nó. Os membros do dicionário são
 - chave: dado do tipo `int`;
 - prox: referência para um dado do tipo `no`;
3. Construa em C++ e Python um programa que:
 - (a) crie 3 nós `n1`, `n2` e `n3`, inicializando os dois últimos(`n2` e `n3`) com chaves distintas e apontadores nulos;
 - (b) armazene um inteiro em `n1`;
 - (c) faça `n1` apontar para `n2`, e `n2` para `n3`;
 - (d) imprima para cada nó sua chave e o endereço do nó para o qual aponta (no Python, utilizar o comando `id()`, para retornar o endereço de memória de uma variável;
4. Construa um programa em Python que
 - (a) Crie diversos nós, utilizando os construtores de dicionário vistos na aula;
 - (b) Aplique as funções de manipulação de dicionários vistas em aula nos nós, mostrando em tela os resultados;

2.2 Parte II

A lista encadeada é uma dos Tipos Abstrados de Dados (TADs) que veremos ao longo do curso. A Figura 2 ilustra esta estrutura de dados. A lista encadeada é caracterizada pelo nó cabeça (*head*) e pelo nó cauda (*tail*). O primeiro corresponde ao início da lista, não sendo apontado por nenhuma outra célula, ao passo que o último é o fim da lista, não apontando para ninguém.

1. Implementar em linguagem C++, a estrutura `ListaEncadeada`. Os membros da estrutura são:
 - `head`: dado do tipo `no`;
 - `tail`: dado do tipo `no`;
 - `tam`: dado do tipo `unsigned int`;

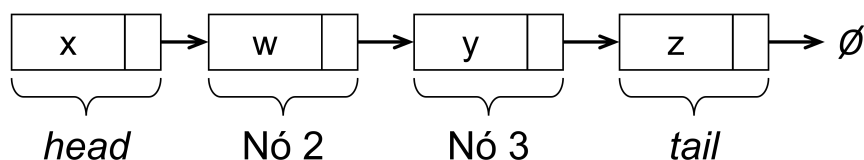


Figura 2: Lista Encadeada

2. Implementar em linguagem Python, uma lista encadeada usando na forma de dicionários com os seguintes itens:
 - head: dado do tipo `no`;
 - tail: dado do tipo `no`;
 - tam: dado do tipo `int`;
3. Construir em ambas as linguagens um subprograma que:
 - (a) Leia o n^o de elementos da lista N ;
 - (b) Leia N chaves em sequência, armazenando-as em N nós;
 - (c) Encadeie os nós, tendo-se por base a ordem de leitura das chaves;
4. Construir em ambas as linguagens um subprograma que imprima a lista encadeada;
5. Construir em ambas as linguagens um subprograma que delete a lista encadeada.
Atenção: O programa deve deletar cada nó da lista, antes deletar a própria lista. No caso do C++, utilizar os operadores `new` e `delete`;
6. Construir em ambas as linguagens um programa que leia, imprima e apague uma lista encadeada;

3 Instruções de realização e entrega

O trabalho deve ser executado de forma individual. Os produtos a serem entregues são:

- Fluxograma dos algoritmos;
- Código-fonte comentado;
- Relatório/resumo executivo (1 ~ 4 páginas), descrevendo como a solução foi alcançada e explicando o funcionamento do algoritmo;

A exceção do código-fonte, entregar os produtos em formato pdf.

3.1 Procedimento de Entrega

- **Data:** Indicada no Classroom;
- **Hora:** 23h:59m - BRT;
- **Local:** Atividade do Google Classroom;